HW1/CS584
Pooya Fayyazsanavi

My submission name on miner : Mr.Nobody
Submission Accuracy : 82%

My Approach:

In this section the approach to final results are discussed thoroughly.

Reading the data:

At first the data converted to pandas data frame with "read_csv()" function. Pandas data frame is a great tool which we can filter and apply functions to our data easily.

Data preprocessing:

- Removing Stop Words:

In the text corpus which we have, there are some words without any meaningful value to our final results. NLTK library is used to get these stop words.

- Stemming:

Stemming is the process of getting the root form of a word. Each word of the data row is passed with a lambda function. SnowballStemmer is used in purpose of stemming.

- Lemmatization:

Lemmatization is the same as stemming with a little difference. Lemmatization ensures that the root word belongs to the language. WordNetLemmatizer is used in purpose of Lemmatization.

- Remove punctuations with RegEx:

Next step is to remove all of the punctuations in our corpus. The characters like [' ," \] can be removed since they don't contribute to the final result.

- Remove numbers:

Numbers can also removed from the text corpus.

- Remove min words:

Usually the words which has few characters is not a meaningful word. For instance, the word oz, kg, I, to name a few.

- Tokenizer:

The process of splitting the sentences to words.

Convert the data to matrix counts:

In order to use the data for our models, the data should converted to a numerical representation. There are different approaches, in this homework assignment, three main approaches were used.

- CountVectorizer
- HashingVectorizer
- TfidfVectorizer

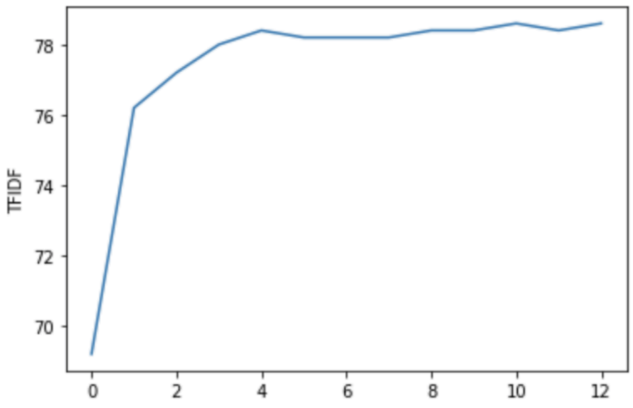
Out of these three methods, Tfidf outperforms the other methods. The method creates from two parts, Term Frequency and Inverse Data Frequency. The L2 normalizer is used to normalize the data.

KNN implementation:

The KNN model is implemented from scratch. It has two parts. "Init()" function initiate the parameters. The "forward" function which gets the test data as parameter and computed the similarity using the Cosin approach. Since KNN has no training process, there is no function implemented for that part. The distance is calculated using matrix multiplication. The majority of votes will determine the output using the K value using the mode function.

Elbow Method:

Different K values were tested in order to find the minimum error rate through our model. The best K for our model is K=201 using the TfidfVectorizer. The accuracy is 82%.



Hyper-parameters:

There are different Hyper-parameters which we can tuned during our training process. For instance, the K value in KNN model. Using the ngram_range for our TfidfVectorizer. The max_features is also an important parameter which is set to 50000 to avoid the computation difficulties.